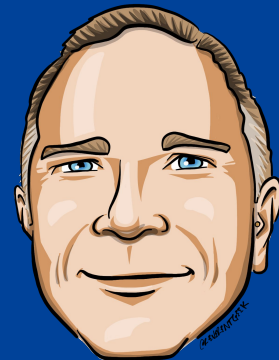
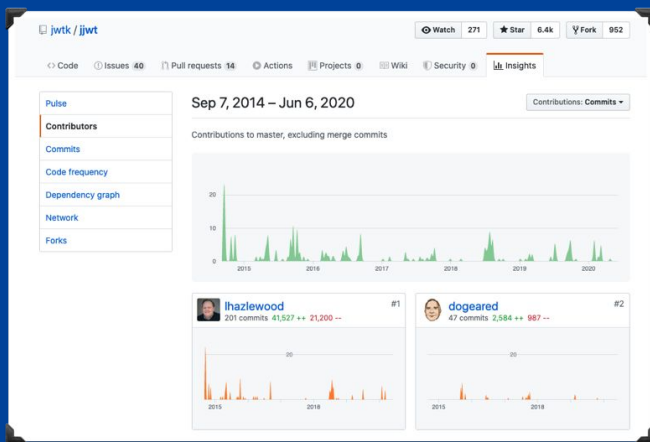
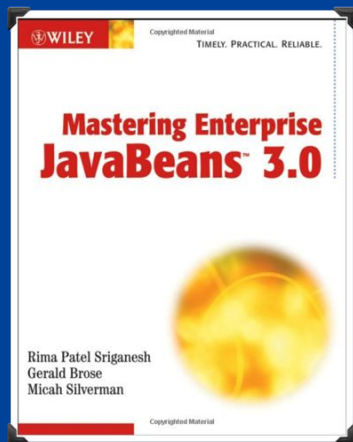


# Hacking OAuth

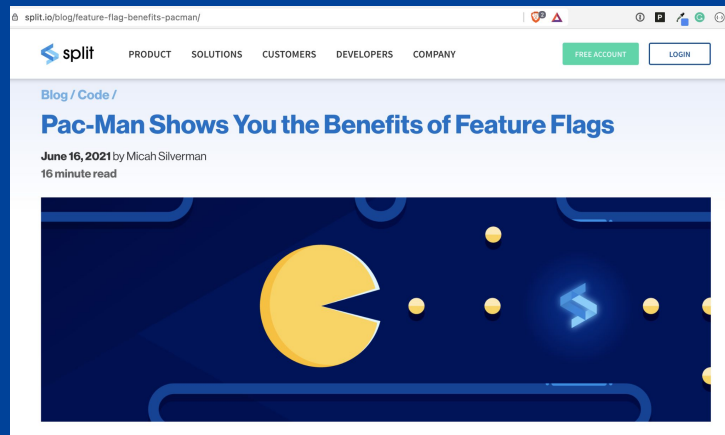
## Pitfalls and Remedies



MICAH SILVERMAN  
Lead Developer Advocate



<https://github.com/jwtk/jjwt>



<https://www.split.io/blog>

# Identity use cases (circa 2007)

01 Simple login – forms and cookies

02 Single sign-on across sites – SAML

03 Mobile app login – ???

04 Delegated authorization – ???

# Don't do it This Way!

(Password Anti-Pattern)

## Are your friends already on Yelp?

Many of your friends may already be here, now you can find out. Just log in and we'll display all your contacts, and you can select which ones to invite! And don't worry, we don't keep your email password or your friends' addresses. We loathe spam, too.

Your Email Service



Your Email Address

(e.g. bob@gmail.com)

Your Gmail Password

(The password you use to log into your Gmail email)

[Skip this step](#)

[Check Contacts](#)

# Don't do it This Way!

(Password Anti-Pattern)

The screenshot shows the 'Step 1: Find Friends' section of Facebook's interface. It features a form for finding friends via email. The form includes a 'Your Email:' field and an 'Email Password:' field. A 'Find Friends' button is located below the password field. A security notice states: 'Facebook will not store your password.' Below the form, there are three options for finding friends: 'Yahoo!', 'Windows Live Hotmail', and 'Other Email Service', each with a 'Find Friends' link.

**Step 1**  
Find Friends

**Step 2**  
Profile Information

**Step 3**  
Profile Picture

**Are your friends already on Facebook?**  
Many of your friends may already be here. Searching your email account is the fastest way to find your friends on Facebook.

**Gmail**

Your Email:

Email Password:

**Find Friends**

Facebook will not store your password.

**Yahoo!** [Find Friends](#)

**Windows Live Hotmail** [Find Friends](#)

**Other Email Service** [Find Friends](#)

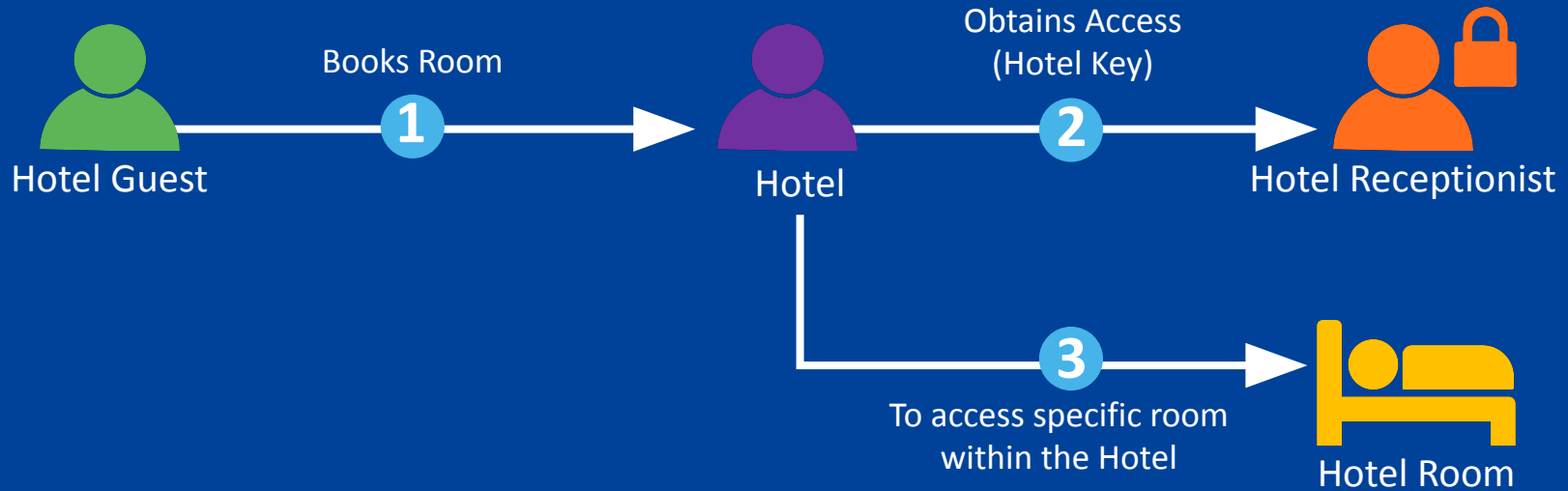
## The Delegated Authorization Problem

HOW CAN I LET A WEBSITE  
ACCESS MY DATA, WITHOUT  
GIVING IT MY PASSWORD?

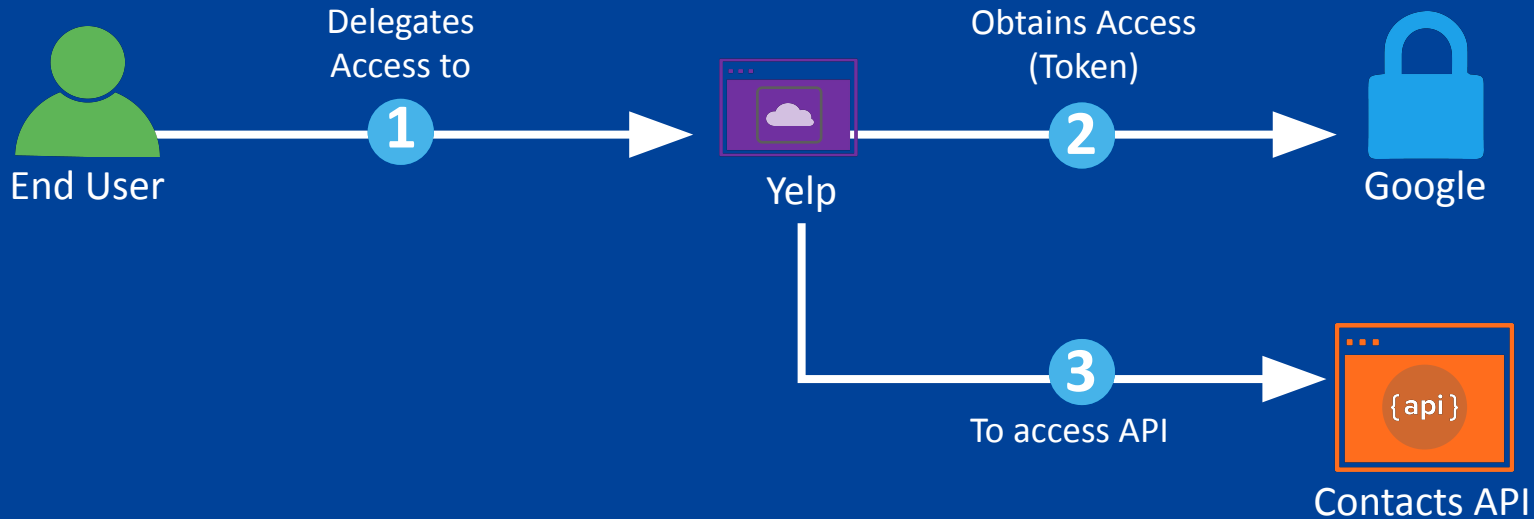
## A lot of confusion around OAuth:

- ⇒ Terminology and jargon
- ⇒ Incorrect advice
- ⇒ Hard to find a life-like example

# The Hotel Analogy



# OAuth 2.0: App Scenario





# OAuth 2.0 Terminology

Hotel	App	OAuth	Description
Hotel Guest	End User	Resource Owner	Wants a Client App to do something on their behalf
Hotel	Yelp	Client Application	Needs authorization to interact with an API on behalf of a user
Hotel Receptionist	Google	Authorization Server	Grants access (in the form of tokens) to an app
Hotel Room	Contacts API	Resource Server	Has an API that an app can use if presented with a token

# Delegated Authorization with OAuth 2.0

I trust Gmail and I kind of trust Yelp. I want Yelp to have access to my contacts only.



yelp.com

Connect with Google

# OAuth 2.0 implicit flow

Client

Yelp Angular app

Connect with Google

Hello!



Resource owner

Go to authorization server

Redirect URI: **yelp.com/callback**

Response type: **token**

Scope: **profile contacts**

Authorization server

accounts.google.com

Email

Password



Request consent  
from resource owner

Back to redirect URI  
with token

Talk to resource server  
with access token

contacts.google.com



accounts.google.com

Allow Yelp to access your public profile and contacts?

No Yes



# DEMO

## Implicit Flow Detector

# Confidential Clients



## **Application running on a server**

Has the ability to keep strings secret  
since code is running in a trusted  
environment

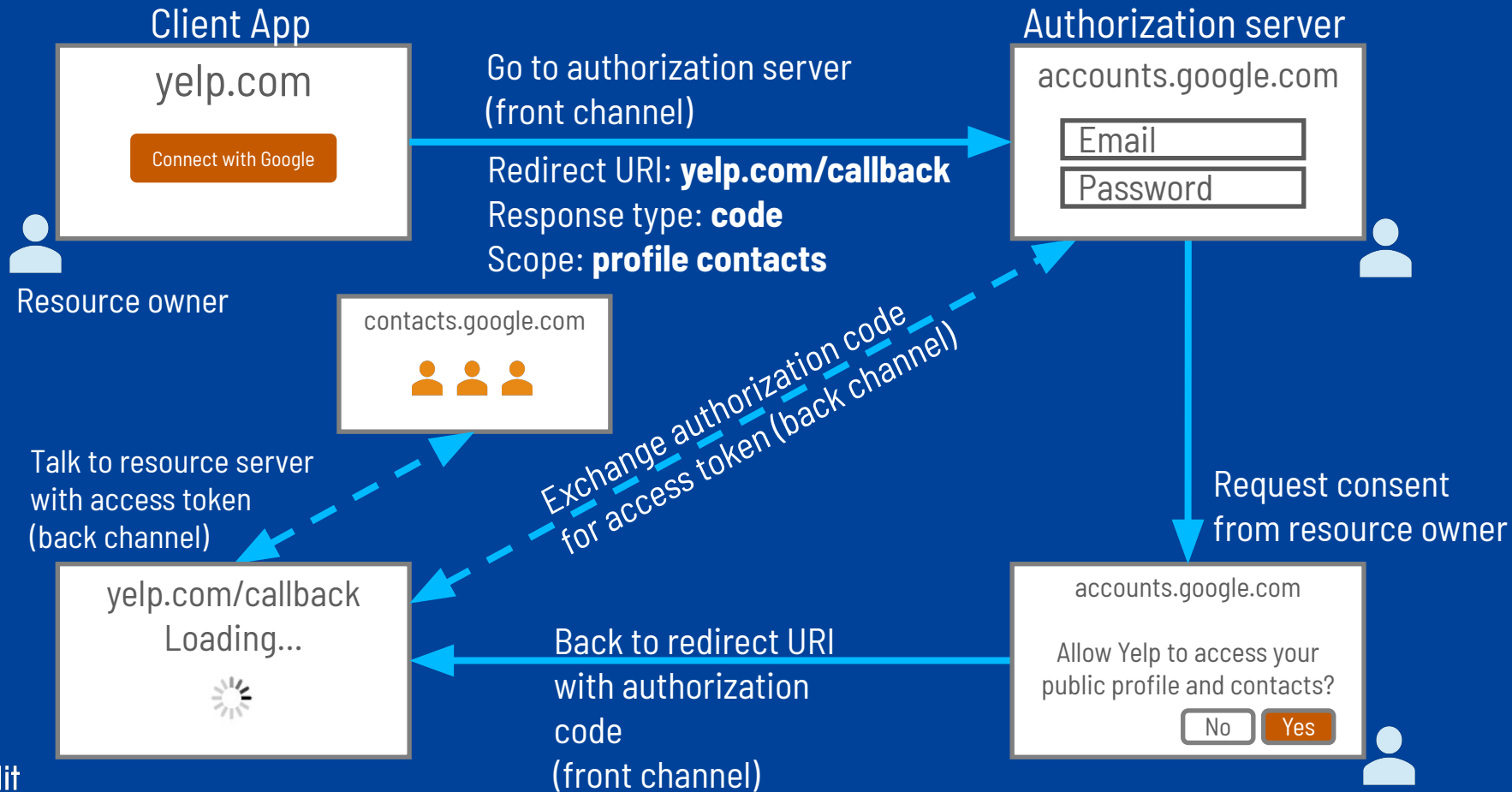
# Public Clients



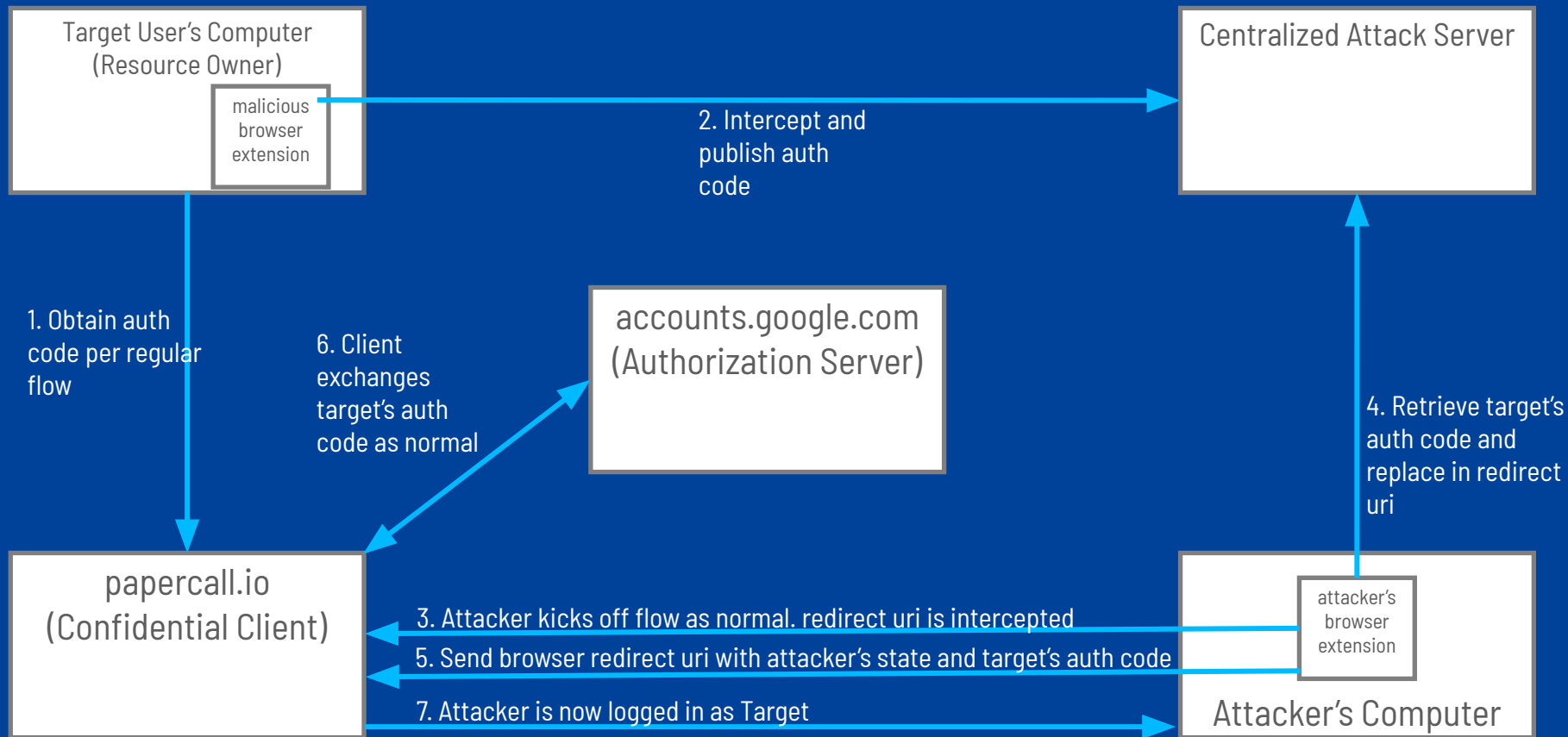
## **The application can't keep strings secret**

Javascript/Single-Page apps: "view source"  
Native apps: decompile and extract strings

# OAuth 2.0 authorization code flow



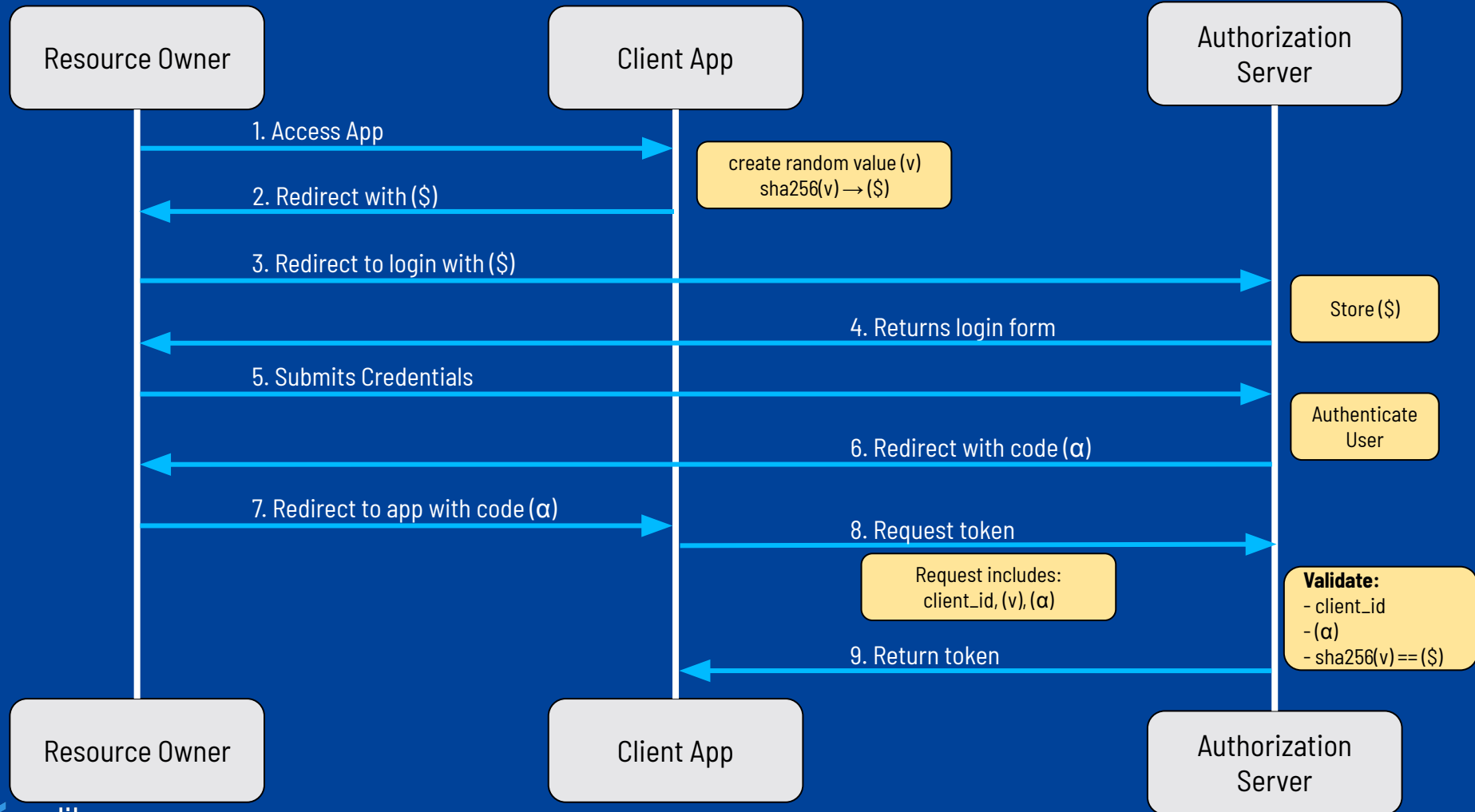
# Authorization Code Injection Attack



# DEMO

## Auth Code Injection



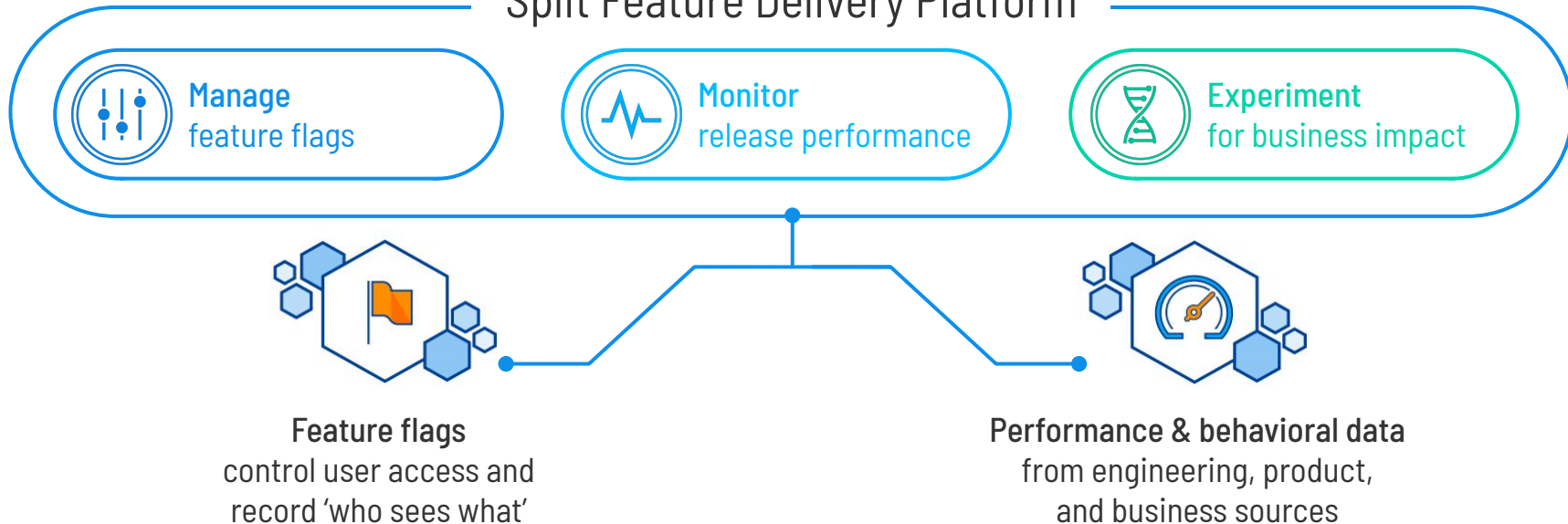


DEMO  
PKCE CLI

# Impact-driven development requires data

Only Split's unified platform closes The Impact Gap

## Split Feature Delivery Platform



## Titanium Sponsors



## Platinum Sponsors



## Gold Sponsors



## Hacking OAuth: Pitfalls and Remedies

Micah Silverman  
@afitnerd

More on OAuth 2.0: **oauth.com**

@splitsoftware

Free developer account: **split.io/signup**

